Hierarchical Module Discovery

Richard A. Watson

Organismic and Evolutionary Biology, Harvard University
2102 Biological Labs., 16 Divinity Avenue, Cambridge, MA 02138, USA.
rwatson@oeb.harvard.edu

Top-down hierarchical or recursive problem decomposition is a familiar and highly successful approach to design and optimization in engineering domains. However, top-down decomposition of a system or problem into more manageable parts requires *a priori* domain knowledge of what kinds of subsystems are feasible and what kinds of subsystem interfaces are manageable. To a large extent this depends on knowledge of existing components that have previously been identified as useful 'building-blocks' in the problem domain. In this paper we overview recent work that has developed an abstract model of automatic module discovery. In an abstract domain, our method discovers modules hierarchically in a bottom-up fashion that allows their re-use in subsequent higher-level structures. This method discovers modules without any *a priori* knowledge of the problem's modular structure. We outline our approach and results.

1. Building-block discovery

Building-block discovery in engineering design

Top-down hierarchical or recursive problem decomposition is a familiar and highly successful approach to design and optimization in engineering domains. However, top-down decomposition requires *a priori* domain knowledge of what kinds of subsystems are feasible and what kinds of subsystem interfaces are manageable. To a large extent this depends on knowledge of existing components that have previously been identified as useful 'building-blocks' in the problem domain.

Over longer timescales, more complex systems can be approached by utilizing assemblies of existing components; and looking backwards, components available off-the-shelf today are themselves assemblies of simpler components identified earlier. This might suggest that over the longer timescale, the ostensibly top-down approach to a given design problem in fact involves relatively limited explorations in assemblies of components that have been incrementally identified bottom-up over the history of the domain. This view of the process, as a non-teleological incremental process of domain knowledge acquisition, is perhaps more naturally amenable to automation than a top-down process. But although it is clear that new systems benefit from domain knowledge extracted and encapsulated from previous exploration in the domain, the details of the processes involved in the creation and identification of new components in engineering methods are not well understood.

Building-block discovery in natural evolution

The theory of evolution by natural selection (Darwin 1859) provided an alternative to the 'top-down' forethought of an intelligent creator and provided an explanation of how limited random variation could be selected and accumulated over long timescales to arrive at complex biological systems in a bottom-up fashion. The traditional view of natural evolutionary processes relies entirely on the incremental accumulation of mutations: more complex biological systems arise (if they do) from the gradual *accretion* of small random variations applied to extant systems. Such a process does not employ encapsulation and re-use of domain knowledge in the production of new variants. In the accretive model of evolution, the variation enabling new systems is of the same form now as it was at the beginning of life – i.e. small random variation arising from replication errors. Accordingly, in this view, there is no sense in which new biological systems are created in a manner that exploits problem decomposition.

However, there are in fact many different variation mechanisms that are in some circumstances better understood as the *composition* of extant systems into a new arrangements rather than the gradual accretion of small random mutations: For example, sexual recombination, hybridisation, allopolyploidy, gene duplication, horizontal gene transfer, endosymbiosis, and symbiogenesis (see Watson 2002). Simply by employing a variation mechanism that re-uses extant genetic material, rather than adding new random genetic material, such compositional mechanisms potentially allow for subsystems to be encapsulated and re-used as 'building-blocks' to bias future variation.

Although the fact of compositional events in natural evolution is clear (Margulis 1970), and such events have been involved in several of the major transitions in evolution (Maynard Smith and Szathmary 1995), the details of the processes that govern the creation of new levels of organization in biological evolution, like their analogues in engineering, are not well understood.

The goal of our research has been to better understand the principles necessary for the successful automatic discovery of re-usable modules or building-blocks in a bottom-up hierarchical fashion. This would both facilitate a better understanding of natural evolutionary processes and of the impact that compositional mechanisms have on evolvability, and also facilitate practical engineering applications in automatic design and optimization.

The remainder of this paper is structured as follows: The next section briefly outlines conceptual issues concerning the definition of a good building-block. Section 3 details an abstract system that we have used to illustrate the concepts involved and assess our automatic module discovery algorithms. Section 4 quantifies the concept of *modular interdependency* using this example system. And Section 5 describes our recently developed module discovery methods. Finally, we discuss the impact of this work in the conclusions.

2. What are building-blocks?

The idea that a subsystem might be re-used in other more complex systems suggests the notion that perhaps a good building-block is defined by the fact that it is useful in many contexts. The extreme extension of this is a building-block whose utility is completely independent of context (e.g. Mitchell and Forrest 1993). However this is too simplistic: When building-blocks are independent of one another, with no inter-module interactions, subsequent higher-level organization becomes degenerate (Watson 2002b). Specifically, if there is no interaction between low-level building-blocks then higher-level building-blocks can, at best, only be arbitrary collections of low-level building-blocks – there is no sense in which the identification of a higher-level building-block could assists in de-coupling dependencies between low-level building-blocks when there are no dependencies between low-level building-blocks.

Simon (1969) suggests a notion of '*nearly*-decomposable' systems that comes some way to providing a usable definition of building-blocks. Simon suggests that a module is a subset of the system where intra-subset dependencies are stronger or more numerous than inter-subset dependencies. This view of modularity is widely adopted. However, although this model clearly includes inter-module interactions, inter-module interactions are considered to be somehow weaker or less significant than intra-module interactions. This leads to problems when these notions are scaled-up over subsequent hierarchical levels – specifically, with each increase in level, the interactions become less significant. This makes it difficult to provide a quantitative example that permits significant interactions at all scales. It is telling that Simon's quantitative examples omit inter-module dependencies altogether (Watson 2002b).

It also appears that something is inadequate in this conception of modularity when we consider a natural example of a module. For example, we generally consider a biological cell to be a meaningful module of a multi-cellular organism. Our general intuition that cells are meaningful entities likely comes from the property of encapsulating internal processes – but we would not generally suppose that (the remaining) inter-cellular interactions are unimportant or insignificant, and we generally admit the possibility that the functioning of the cell is *strongly* dependent on its environment.

This leaves us with a need to define a quantitative model of a modular system that has clearly identifiable modules, such that good configurations of a module are useful in many contexts, yet, the identification of a good configuration should not be independent of the state of other modules, and ideally this modularity should appear consistently through many hierarchical levels. To understand this better we define a simple hierarchically modular system that exhibits these properties.

3. Modular Interdependency

In this section we clarify the ideas introduced above with the use of a simple example dynamical system. We show that structural modularity in the connectivity of the system's components does not imply isolation, or near independence, of the dynamical behavior of modules. Thus we show that there is a meaningful sense in which a system may be modular and yet have significant interdependencies between modules at all scales. This leads us to define the concept of *modular interdependency*.

An example modular dynamical system

Let us consider the domain of gene regulation networks as an example dynamical system. We suppose for simplicity, that each gene may be in one of two states: "high", meaning highly expressed, or otherwise, "low". Let the future state of each gene be determined by some function of the states of the genes that regulate its expression. (In general, this describes a 'Random Boolean Network', Kauffman 1993).

We can use a graph to represent the connectivity of regulation activity where nodes represent genes, and edges represent regulation interactions. In general, edges may be directed, but, for our purposes here, we can simplify our model by asserting that all interactions are bi-directional - i.e. an edge between A and B means that A regulates B, and B regulates A. It also suits our purposes to assume that all nodes have a self-recurrent connection so that a node's future state is partly determined by its own state at the previous time step.

To define a modular network, we modify a fully connected network such that the interactions within particular subsets of genes will be stronger than the interactions between genes in different subsets. To represent this we will simply use multiple edges between genes (since we do not really wish to distinguish numerous interactions from strong interactions). Figure 1, thus shows an example system that exhibits a clear, two-module structure.

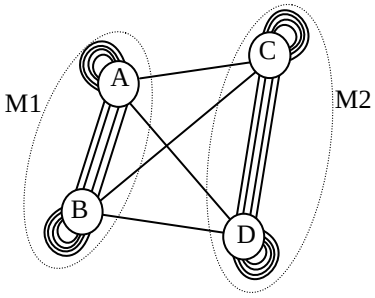

Figure 1: An example system of 4 genes, A, B, C, and D, arranged in two modules, M1 and M2, where intra-module (and self-recurrent) interactions are of strength 4, and inter-module interactions are of strength 1.

This graph describes the structural properties of our system. But as we have noted, we wish to examine its dynamical properties and the structural properties alone are insufficient to make conclusions about dynamical properties. For illustration we use a simple 'voting' style discrete-time update rule. Specifically, the probability of a gene taking a given state in the next time step will be equal to the proportion of regulating connections from genes that are themselves in that state at the current time step. i.e.

Let "high"=1 and "low"=0 then,

$$P(A_{t+1}=1)=\tfrac{1}{K_A}\sum_{i=1}^{k_A} Bi_t \,,$$

$$P(A_{t+1}=0)=1-P(A_{t+1}=1). \qquad \text{Eq.1.}$$

where A_t is the state of gene A at time t, k_A is the number of edges connecting to A, B_i is the state of the i^{th} regulating gene connected to A. This update rule is popular in models of physical dynamical systems such as Ising models.

We have now defined the structure and the dynamics of the system. For some purposes, the structural modularity of a system may be sufficient, but if we are interested in functional modularity that may be determined by dynamical properties such as the stability and location of attractors, then we must provide further analysis. The long term (rather than immediate) effects of changes in one module on state changes in another module can be characterized by changes in the attractors of a system. In general, if the attractors of a subsystem or module are unaffected by state changes in another module then this is an important form of independence. At the opposite extreme, if we find that the attractors of a subsystem are completely different this indicates an important form of inter-module dependency.

By way of example, let us analyze our example system by examining which configurations of M1 are most stable, and how these configurations differ with state changes in M2. For these purposes it is sufficient to use a simple measure of stability: specifically, let us define the *stability* of a network, or subnetwork, as the probability that no gene in the network changes state. (This definition of stability closely parallels the *free-energy* measure of a configuration in an Ising model.[1]) The stability of a network is therefore the product of the stability for each gene in the network. For example, in a system of two genes, A and B, where the probability of A remaining in the same state is S(A), and the probability of B remaining in the same state is S(B), the stability of the AB system is simply S(A)S(B).

Clearly, since there are inter-module edges, the stability of a given configuration is sensitive to the state of genes in the other module, and the configuration which is most stable may also differ.

Let us write the states of the 4 genes in our system as A, B, C, and D; and we will write the stability of a subset of variables *x*, given the state of a subset of variables *y*, as S(*x*, *y*): e.g. the stability of M1, given the state of M2, is written as S(AB,CD). Given the symmetries in our example system, the effect of 01 and 10 are the same, so we will list only the combinations 00, 01, and 11 for each module. Then, for our example system, using Eq.1:

A B C D	S(A,BCD)	S(B,ACD)	S(AB,CD)
0 0 0 0	10/10	10/10	1
0 1 0 0	6/10	4/10	0.24
1 1 0 0	8/10	8/10	0.64
0 0 0 1	9/10	9/10	0.81
0 1 0 1	5/10	5/10	0.25
1 1 0 1	9/10	9/10	0.81
0 0 1 1	8/10	8/10	0.64
0 1 1 1	4/10	6/10	0.24
1 1 1 1	10/10	10/10	1

We immediately notice that the configuration of M1 that is most stable is strongly sensitive to the configuration of states in M2. Specifically, the most stable configuration of M1 when CD=00 is AB=00, and the most stable configuration of M1 when CD=11 is AB=11. The systems' symmetries provide corresponding stabilities for M2.

This example system is therefore sufficient to illustrate the following: 1) This system is *structurally* modular in the sense that it has stronger (more numerous) intra-module dependencies than inter-module dependencies, but it has strong *functional* inter-module dependencies in the sense that a simple dynamical property – the most stable configuration for a module – is strongly dependent on the state of other modules. 2) Following from this, if the function of a system depends on dynamical properties, then we cannot assume that module interactions are unimportant

[1] Our observations of stabilities and attractors in this example system will be unsurprising for those readers familiar with Ising models and their dynamics, but working through the example in some detail is useful for providing clarity.

even if the structural modularity of the system is unambiguous.

4. Analyzing Modular Interdependency

From the discussion of our example system in the previous section one might conclude that (despite the apparent structural modularity) this system is in fact not modular in respect to the property of 'most stable configuration'. But, it should also be noted that even though the configuration of a module that is most stable is strongly dependent on the state of the other module, there is some degree of independence in this respect also. Specifically, we notice that although the most stable configuration of states may be either 00 or 11 it is always one of these and never 01 or 10 regardless of the state of the other module. This means that there is something we know about the property of interest, the most stable configuration, that is *independent* of inter-module interactions.

From these observations we arrive at a means to quantify the independence of a module, given a property of interest. Specifically, given some property that identifies particular configurations, like the configuration that is most stable, we can assess what we know about the identity of such states that is independent of the state of other modules. In this example, there are four possible configurations for a module, C=4. Accordingly, a meaningful definition of *non*-decomposability, given this property of the system, is that we know nothing about which of the four possible states is the most stable state. This will be the case when, for every configuration of M1, there is some configuration of M2 (or the remainder of the system) that would make that configuration of M1 the most stable. However, in our example system, this is not the case. There are only two configurations that could be maximally stable, C'=2; regardless of the state of the other module. In our modular system $1<C'<C$. In a non-decomposable system, C'=C.

In general, we will say that when $C'<C$ the system is *decomposable*. But note that there is a special case of decomposability where C'=1 which means that the configuration of interest is always the same regardless of the state of the remainder of the system. In this case, the module is fully independent in the property of interest, a case which we call *separable*. In the general case where a system is decomposable but not separable, i.e. $1<C'<C$, as in our example, we say the system exhibits *modular interdependency*.

This manner of quantifying the dependence or independence of one module from another, given a property of interest, is detailed in (Watson 2002). In that work, the property of interest is the configuration that maximizes the fitness of the system. This understanding of modularity clearly allows meaningful decomposability when $C'<C$, and yet still allows the possibility of important interactions between modules when $C'>1$.

Modular interdependency may be exhibited in a system at numerous hierarchical levels. We can simply expand the network shown in Figure 1 so that nodes are clustered into groups and sub-groups recursively. Alternatively, we may abstract the stability function into a simple binary function over strings of length $N=2^P$, where P is the number of hierarchical levels in the system (see Watson and Pollack 1998, and Watson 2001, for details).

The 'evolvability' of systems with modular interdependency

In large, multi-level systems exhibiting modular interdependency it is very difficult for an accretive process to resolve the dependencies between large modules as is necessary to make continued improvement. The interdependencies between modules can require all the states of a module to change at the same time to a specific different configuration. The likelihood of such a change under random modifications decreases exponentially with the size of the module. This corresponds to the likelihood of improving a complex engineering design, like an automobile, by making random changes in the primitive variables, like nuts and bolts and the shape of primitive parts.

However, compositional processes can potentially exploit the modular decomposability of such systems to provide continued improvement. In such cases, a method that is able to 'swap out' a particular high-level component and 'swap in' an alternative high-level component has a better chance of finding improvement than a method that makes random low-level changes. For example, if a disc braking system can be exchanged for a drum braking system then many low-level components change, all at once, to a new specific configuration. Naturally, although this is familiar and obvious in engineering practice, the difficult part of automating such hierarchical compositional processes will be the automatic discovery of appropriate building-blocks. In the next section we describe our methods for achieving this.

5. Modeling building-block discovery

Our basic approach to the hierarchical discovery and re-use of building-blocks is to provide an evolutionary algorithm with the capability of representing modules and provide a selection scheme that favors useful modules. Holland (1975) suggested, at the outset of evolutionary algorithm research, that the exchange of genetic material under sexual recombination may provide the opportunity to represent and manipulate building-blocks. This would come about by virtue of the increased probability of nearby genes to travel together during crossover. Thus sets of co-adapted genes could be exchanged between individuals. Using a hierarchical building-block function based on modular interdependency we were able to show circumstances under which crossover could discover high fitness configurations in time less than order $N^2 \lg^2 N$ (where N is the number of problem variables), whereas no mutation based algorithm can optimize this function in time less than $2^{N/2}$ (Watson

2001). However, this result was dependent on, amongst other things, the close correspondence of genetic linkage (the position of variables on the chromosome) with (epistatic) dependencies between problem variables. Accordingly, this model omits a critical aspect of building-block discovery – specifically, the problem of determining which subsets of variables constitute a building-block – because this knowledge is assumed in order to provide an appropriate ordering of genes on the chromosome.

To overcome this limitation we took further inspiration provided by biological systems and modeled a process of symbiotic encapsulation. Sexual recombination and symbiotic encapsulation are both compositional processes but whereas sexual recombination exchanges parts of genomes between closely related individuals, symbiotic encapsulation essentially joins together whole genomes and may occur between unrelated species. Our symbiotic encapsulation model uses variable-size chromosomes that explicitly represent subsets of strongly interdependent genes/problem variables.[2] An (endosymbiotic) join operator then creates new sets of genes from collections of existing sets.

Initial studies found that in addition to providing these mechanisms to represent and manipulate subsets of variables, it was necessary to provide a new selection model. A straightforward selection model using the average performance of a partially specified genome (corresponding to 'schema fitness' – Holland 1975) allowed 'bloated' sub-optimal modules to take over the population (Watson 2002).

The alternative selection method we developed uses Pareto domination to compare two proposed modules instead of average performance. Using Pareto domination, a module may competitively exclude another module only if it is better than the alternate module in *all* contexts (actually, if it is at least as good as the alternate module in all contexts and better in at least one context). This is much more conservative than the normal averaging method where a module may competitively exclude another if it is better *on average* over all contexts. The Pareto domination selection method allows alternate modules to persist in the population because they have some utility in some contexts even if they are not good on average. As an evolutionary process, the selection model can be understood as the maintenance of different types in an coevolutionary ecosystem where competitive exclusion only occurs when one species outperforms another in *all* 'niches' (rather than when one outperforms the other *on average* over all niches).

The Symbiotic Evolutionary Adaptation Model, SEAM (Watson & Pollack 2001, Watson 2002), based on this Pareto dominance selection method, is able to discover and manipulate modules in the problem domain effectively. It discovers optimal solutions in polynomial time, without using any *a priori* information about which subsets of variables form a building-block.

SEAM is effective because the Pareto dominance selection method properly corresponds with the concept of modularity we defined above. The average performance selection method will only provide optimal solutions when the configuration for a module is independent of context – i.e. when modules are separable, C'=1. In contrast, the Pareto dominance selection method maintains modules if there is some context for which they are useful. If C'=C this method will try to maintain all possible configurations that each module could take because there will be some context where they are needed. This will prevent progress. But when C'<C some configurations for a proposed module will be inferior to alternatives in all contexts. In this case, such possibilities can be eliminated from further consideration. In this manner, useful modules can be retained and useless modules can be discarded. Accordingly, SEAM works well in systems with modular interdependency, i.e. where C>C'>1.

6. Conclusions

Our work on the concept of modularity and a definition of building-blocks indicates that simplistic concepts of modularity can be misleading. Specifically, the notion that building-blocks are separable precludes meaningful hierarchy altogether, and the notion that they are 'nearly' separable prevents scale-invariant hierarchy. Our example system in Section 3 is used to illustrate a more general class of modular systems based on *modular interdependency*. This concept is concrete enough that we can quantify how decomposable a module is with respect to its interaction with the remainder of the system. This quantification is simply a count of the number of module configurations that maximize some criterion of interest over all possible contexts. Although this may be difficult to calculate in non-trivial real systems it is an explicit and precise notion of decomposability. Moreover, it is a concept of modularity that allows meaningful modules without assuming that modules are separable or nearly separable and this in turn permits meaningful modularity at all scales in a system.

In the automatic discovery of building-blocks it is necessary to respect the meaning of a building-block appropriately. For example, methods that reward a module because it is good on average will only provide optimal solutions in the special case where modules are separable. However, our method of rewarding modules based on Pareto dominance respects the fact that the optimal configuration for a module may be different in different contexts, and this method thereby maintains alternate configurations of a module appropriately.

Ongoing work seeks to make our methods more robust for application in practical engineering domains, and in parallel, to increase our understanding of analogous biological processes.

[2] In contrast, sexual recombination exchanges subsets of genetic material taken from full-length genotypes, and this represents building-blocks only implicitly using the location of genes on the chromosome.

References

Darwin, C, 1859, *On the Origin of Species by Means of Natural Selection*, John Murray, London.

Holland, JH, 1975, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: The University of Michigan Press.

Kauffman, SA, 1993, *The Origins of Order*, Oxford University Press.

Margulis, L, 1970, *Origin of eukaryotic cells*. Yale University Press, New Haven.

Maynard Smith, JM & Szathmary, E, 1995 *The Major Transitions in Evolution*, WH Freeman.

Mitchell, M, Forrest, S, and Holland, JH, 1992, "The royal road for genetic algorithms: Fitness landscapes and GA performance", in *First ECAL*. Cambridge, MA: MIT Press, pp. 245-54.

Simon, HA, 1969, *The Sciences of the Artificial*, Cambridge, MA. MIT Press.

Watson, RA, 2001, "Analysis of Recombinative Algorithms on a Non-Separable Building block Problem", *FOGA VI*, (2000), Morgan Kaufmann, pp. 69-89.

Watson, RA, 2002, *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis*, PhD Dissertation, May 2002, Computer Science Dept., Brandeis University, MA, USA.

Watson, RA, 2002b, "Modular Interdependency in Complex Dynamical Systems", *Proceedings of The Workshop on Dynamical Hierarchies* at Artificial Life 8, Sydney, Australia, December 2002.

Watson, RA, & Pollack, JB, 2001, "A Computational Model of Symbiotic Composition in Evolutionary Transitions", *BioSystems*, to appear 2003.

Watson, RA, Hornby, GS, & Pollack, JB, 1998, "Modeling Building block Interdependency", *Procs. of PPSN V*, Springer. pp.97-106.